



AF/AT
IFW

PATENT
Customer No. 22,852
Attorney Docket No. 06502.0396

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:)
Paul A. LOVVIK et al.) Group Art Unit: 2164
Application No.: 10/051,277) Examiner: S. R Pannala
Filed: January 22, 2002)
For: METHOD AND APPARATUS) Confirmation No.: 7605
FOR PROCESSING A)
STREAMED ZIP FILE)

Mail Stop Appeal Brief--Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

TRANSMITTAL OF APPEAL BRIEF (37 C.F.R. 41.37)

Transmitted herewith is the APPEAL BRIEF in this application with respect to the
Notice of Appeal filed on September 30, 2005.

This application is on behalf of

Small Entity Large Entity

Pursuant to 37 C.F.R. 41.20(b)(2), the fee for filing the Appeal Brief is:

\$250.00 (Small Entity)
 \$500.00 (Large Entity)

TOTAL FEE DUE:

Appeal Brief Fee	\$500.00
Extension Fee (if any)	\$120.00

PATENT
Customer No. 22,852
Attorney Docket No. 06502.0396

Total Fee Due \$620.00

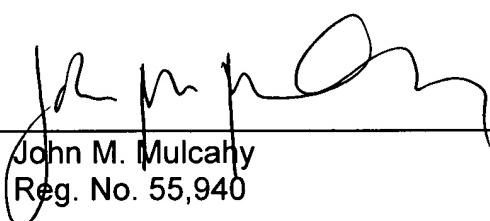
Enclosed is a check for \$620.00 to cover the above fees.

PETITION FOR EXTENSION. If any extension of time is necessary for the filing of this Appeal Brief, and such extension has not otherwise been requested, such an extension is hereby requested, and the Commissioner is authorized to charge necessary fees for such an extension to our Deposit Account No. 06-0916. A duplicate copy of this paper is enclosed for use in charging the deposit account.

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: December 29, 2005

By: _____


John M. Mulcahy
Reg. No. 55,940



PATENT
Customer No. 22,852
Attorney Docket No. 06502.0396-00

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:)
Paul A. LOVVIK et al.) Group Art Unit: 2164
Application No.: 10/051,277) Examiner: S. R. Pannala
Filed: January 22, 2002)
For: METHOD AND APPARATUS) Confirmation No.: 7605
FOR PROCESSING A)
STREAMED ZIP FILE)

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

APPEAL BRIEF UNDER BOARD RULE § 41.37

In support of the Notice of Appeal filed September 30, 2005, and further to Board Rule 41.37, Appellants present this brief and enclose herewith a check for the fee of \$500.00 required under 37 C.F.R. § 1.17(c).

This Appeal Brief is being filed concurrently with a petition for an Extension of Time for one month, and the appropriate fee.

This Appeal responds to the final rejection of claims 1-19 (all currently pending claims) in the Office Action dated June 30, 2005.

12/30/2005 JADD01 00000054 10051277

01 FC:1402
02 FC:1251

500.00 OP
120.00 OP

If any additional fees are required or if the enclosed payment is insufficient, Appellants request that the required fees be charged to Deposit Account No. 06-0916.



TABLE OF CONTENTS

I.	REAL PARTY IN INTEREST	1
II.	RELATED APPEALS AND INTERFERENCES	2
III.	STATUS OF CLAIMS	3
IV.	STATUS OF AMENDMENTS	4
V.	SUMMARY OF CLAIMED SUBJECT MATTER.....	5
	A. Independent claim 1	5
	B. Independent claim 5	5
	C. Independent claim 11	6
	D. Independent claim 17	7
VI.	GROUNDSS OF REJECTION TO BE REVIEWED	8
VII.	ARGUMENT	9
	A. The rejection of independent claim 1 under 35 U.S.C. § 103(a) must be reversed because Hendl et al. and Basin et al. fail to support the rejection of claim 1, which recites, among other things, “receiving a stream of data containing an un-extracted zip file, wherein the zip file comprises a set of files and a central directory; and enabling a process to access contents of the central directory as it is received.”	9
	B. The rejection of dependent claims 2-4 under 35 U.S.C. § 103(a) must be reversed for at least the same reasons given above with respect to claim 1.	11

C. The rejection of independent claims 5 and 11 under 35 U.S.C. § 103(a) must be reversed because Hendler et al. and Basin et al. fail to support the rejection of these claims, which recite, among other things, a receiving module “for receiving a streamed un-extracted zip file,” and an interface module “for accessing contents of a central directory of the streamed zip file as the central directory is received.”	12
D. The rejection of dependent claims 6 and 12 under 35 U.S.C. § 103(a) must be reversed because Hendler et al. and Basin et al. fail to teach that “the interface module is a Java class comprising a central header subclass and a central directory subclass.”	14
E. The rejection of dependent claims 7-10 and 13-16 under 35 U.S.C. § 103(a) must be reversed for at least the same reasons given above with respect to claims 6 and 12.	15
F. The rejection of independent claim 17 under 35 U.S.C. § 103(a) must be reversed because Hendler et al. and Basin et al. fail to support the rejection of claim 17, which recites, among other things, “an interface ... for use with a receiver configured for receiving a streamed un-extracted zip file, wherein the zip file comprises a set of files and a central directory, the interface comprising a process for accessing contents of the central directory as it is received.”.....	16

G. The rejection of dependent claims 18 and 19 under 35 U.S.C. § 103(a) must be reversed for at least the same reasons given above with respect to claim 17	17
H. Conclusion.....	18
VIII. CLAIMS APPENDIX	19
IX. EVIDENCE APPENDIX	23
X. RELATED PROCEEDINGS APPENDIX.....	24



I. REAL PARTY IN INTEREST

The real party in interest is Sun Microsystems, Inc., the assignee of the entire right, title and interest in the present Application.

II. RELATED APPEALS AND INTERFERENCES

There are currently no other appeals or interferences, of which Appellants, Appellants' legal representative, or Assignee are aware, that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-19 remain pending. No claims have been cancelled. The Final Rejection of claims 1-19 is appealed.

IV. STATUS OF AMENDMENTS

An Amendment After Final was filed August 30, 2005. In the Advisory Action dated September 22, 2005, the Examiner indicated that this Amendment would not be entered for purposes of appeal.

V. SUMMARY OF CLAIMED SUBJECT MATTER

In accordance with Rule 41.37(c)(1)(v), Appellants present a concise explanation of the subject matter defined in each of the independent claims involved in the appeal, referring to the specification by page and line number and the drawings by reference characters.

A. Independent claim 1

Independent claim 1 is directed to a method of accessing a streamed zip file (see, e.g., p. 2, ¶ 002, l. 1, through p. 3, l. 9; and FIG. 1, reference character 100). The method comprises: receiving a stream of data containing an un-extracted zip file (see, e.g., p. 11, ¶ 047, ll. 2-3; and p. 11, ¶ 048, ll. 13-15), wherein the zip file comprises a set of files (see, e.g., p. 7, ¶ 019, ll. 2-4; and FIG. 1, reference character 102) and a central directory (see, e.g., p. 7, ¶ 019, ll. 4-6; and FIG. 1, reference character 104); and enabling a process to access contents of the central directory as it is received (see, e.g., p. 12, ¶ 049, ll. 11-13; p. 17, ¶ 052, ll. 1-3; and FIG. 3, reference character 310).

B. Independent claim 5

Independent claim 5 is directed to a system (see, e.g., p. 10, ¶ 046, ll. 1-2; FIG. 2, reference character 200; p. 13, ll. 6-7; and FIG. 4, reference character 400) for receiving a streamed zip file (see, e.g., p. 2, ¶ 002, l. 1, through p. 3, l. 9; and FIG. 1, reference character 100), wherein the zip file comprises a set of files (see, e.g., p. 7, ¶ 019, ll. 2-4; and FIG. 1, reference character 102) and a central directory (see, e.g., p. 7, ¶ 019, ll. 4-6; and FIG. 1, reference character 104). The system comprises: a central processing unit (see, e.g., p. 10, ¶ 046, l. 3; FIG. 2, reference character 202;

p. 13, I. 8; and FIG. 4, reference character 402); an application program (see, e.g., p. 11, ¶ 048, II. 1-3; FIG. 2, reference character 214; p. 14, ¶ 051, II. 4-6 ; and FIG. 4, reference character 414) configured for execution by the central processing unit; a receiving module (see, e.g., FIG. 2, reference character 216; and FIG. 4, reference character 420), initiated by the application program, for receiving a streamed un-extracted zip file (see, e.g., p. 11, ¶ 048, II. 13-15; p. 14, ¶ 051, II. 1-10); and an interface module (see, e.g., FIG. 2, reference character 218; and FIG. 4, reference character 422), initiated by the application program, for accessing contents of a central directory of the streamed zip file as the central directory is received (see, e.g., p. 12, ¶ 049, II. 11-13; p. 17, ¶ 052, II. 1-3; and FIG. 3, reference character 310).

C. Independent claim 11

Independent claim 11 is directed to a computer program product (see, e.g., p. 17, ¶ 053) for use in conjunction with a computer system (see, e.g., p. 10, ¶ 046, II. 1-2; FIG. 2, reference character 200; p. 13, II. 6-7; and FIG. 4, reference character 400). The computer program product comprises a computer readable storage medium and a computer program mechanism embedded therein (see, e.g., p. 17, ¶ 053). The computer program mechanism comprises: a receiving module (see, e.g., FIG. 2, reference character 216; and FIG. 4, reference character 420) for receiving a streamed un-extracted zip file (see, e.g., p. 11, ¶ 048, II. 13-15; p. 14, ¶ 051, II. 1-10); and an interface module (see, e.g., FIG. 2, reference character 218; and FIG. 4, reference character 422) for accessing contents of a central directory of the streamed zip file as

the central directory is received (see, e.g., p. 12, ¶ 049, ll. 11-13; p. 17, ¶ 052, ll. 1-3; and FIG. 3, reference character 310).

D. Independent claim 17

Independent claim 17 is directed to a memory (see, e.g., p. 11, ¶ 047; FIG. 2, reference character 210; p. 13, ¶ 050, ll. 1-5; and FIG. 4, reference character 410) for storing data for access by an application program (see, e.g., p. 11, ¶ 048, ll. 1-3; FIG. 2, reference character 214; p. 14, ¶ 051, ll. 4-6 ; and FIG. 4, reference character 414) being executed on a computer system (see, e.g., p. 10, ¶ 046, ll. 1-2; FIG. 2, reference character 200; p. 13, ll. 6-7; and FIG. 4, reference character 400), comprising: an interface (see, e.g., FIG. 2, reference character 218) stored in the memory, the interface for use with a receiver (FIG. 2, reference character 216) configured for receiving a streamed un-extracted zip file (see, e.g., p. 2, ¶ 002, l. 1, through p. 3, l. 9; and FIG. 1, reference character 100), wherein the zip file comprises a set of files (see, e.g., p. 7, ¶ 019, ll. 2-4; and FIG. 1, reference character 102) and a central directory (see, e.g., p. 7, ¶ 019, ll. 4-6; and FIG. 1, reference character 104), the interface comprising a process for accessing contents of the central directory as it is received (see, e.g., p. 12, ¶ 049, ll. 11-13; p. 17, ¶ 052, ll. 1-3; and FIG. 3, reference character 310).

VI. GROUNDS OF REJECTION TO BE REVIEWED

Claims 1-19 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Hundler et al. (U.S. Patent Application Publication No. 2002/0042833 A1) in view of Basin et al. (U.S. Patent Application Publication No. 2002/0120639 A1). The rejections of claims 1-19 are appealed.

VII. ARGUMENT

Claims 1-19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Hundler et al. in view of Basin et al. Appellants respectfully submit that the rejection of each of claims 1-19 must be reversed for at least the following reasons.

- A. **The rejection of independent claim 1 under 35 U.S.C. § 103(a) must be reversed because Hundler et al. and Basin et al. fail to support the rejection of claim 1, which recites, among other things, “receiving a stream of data containing an un-extracted zip file, wherein the zip file comprises a set of files and a central directory; and enabling a process to access contents of the central directory as it is received.”**

Hundler et al. discloses:

[A] method of streaming an archive file (such as a Java Archive [JAR] file) from a server to a client device. The method includes *extracting* files from a Java Archive and streaming the *extracted* files from the server to the client device, receiving the streamed files and storing the received files for access by a Java application.

Hundler et al., ¶ 0009 (emphasis added); see also id., ¶¶ 0067, 0073-76. Thus, as the Examiner admits (see Final Office Action, p. 3, § 5, II. 4-5), Handler et al. fails to teach “receiving a stream of data containing an *un-extracted* zip file,” (Claim 1, I. 2 (emphasis added)).

In an attempt to cure this deficiency, the Examiner cites Basin et al. as teaching the existence of an “‘un-extracted zip file’ as these days almost any file one downloads from the Internet is compressed (un-extracted) in some way,” and asserts that “it would have been obvious to one of ordinary skill ... to have combined the teachings of [Handler et al. and Basin et al.] because Basin’s teachings would have allowed

Handler's method with an easy management and manipulation of archive files." Final Office Action, p. 3, ll. 5-7 and 11-15. Appellants respectfully disagree.

The Examiner relies on Handler et al. as teaching "the claimed step of 'enabling a process to access the contents of the central directory as it is received' as the contents of each local file header is repeated in a central directory 640 located at the end of the zip archive [600]." Final Office Action, p. 4, ll. 4-7 (citing Handler et al., FIG. 6, and ¶ 0069). However, in the Handler et al. method, the JAR (ZIP) file 600 is streamed "as a series of separate modules by *extracting* the ZIP central directory 640, as well as the individual files and their associated headers 631-636, and streaming each of these [extracted] elements 631-636, 640 separately." Handler et al., ¶ 0073, ll. 2-7 (emphasis added).

Consequently, in Handler et al., the central directory record 640 is streamed, and therefore received, *separately*, and in *extracted* form. Id.; see also id., ¶¶ 0067, and 0073-76. In the method of claim 1, the zip file, which "comprises a set of files and a central directory," (claim 1, ll. 2-3) is received in "un-extracted" form (id., l. 2). Thus, contrary to the Examiner's assertions, Handler et al. does not "enable a process to access contents of the central directory as *it is received*," because the process by which Handler et al. accesses the contents of the central directory requires that the central directory to be extracted prior to receipt. See id., ¶ 0073, 0074 and 0076.

Moreover, Basin et al. teaches that "[g]enerally, the contents of a compressed file cannot be accessed unless the archive is uncompressed." Basin et al., ¶ 0004, ll. 1-2. Thus, Basin et al. is not relied upon to teach, and does not teach, "enabling a process to access contents of the central directory as *it is received*." Claim 1, ll. 4-5 (emphasis

added). Instead, Basin et al. teaches that “[w]hen files are dropped [(i.e., downloaded)] from archive to Explorer, Explorer requests available standard data formats. In this instance, the data object will need to uncompress the data.” Basin et al., ¶ 0032, ll. 6-9. In order to extract individual files and/or folders archived in a zip file, “a user opens the zip file in Explorer … and invokes the extract dialog, by selecting the Extract menu item in the right-click context menu. Alternatively, a user may select PKZIP|Extract Here to extract the contents of the archive into the directory where the zip archive resides.” Id., ¶ 0037, ll. 1-8. Thus, the Basin et al. method would allow the central directory to be accessed only *after* the entire zip file has been received and extracted by the user. Id.

Consequently, even if the artisan would have been motivated to modify Handler et al. by streaming the JAR file in un-extracted form (which Appellants dispute), neither Handler et al., nor Basin et al., nor their combination, teach a process for accessing the central directory “as it is received.” For at least these reasons, Handler et al. and Basin et al. fail to support the Examiner’s rejection of claim 1, whether taken alone or in combination. Accordingly, Appellants respectfully request that the rejection of claim 1 under 35 U.S.C. § 103(a) be reversed and the claim allowed.

B. The rejection of dependent claims 2-4 under 35 U.S.C. § 103(a) must be reversed for at least the same reasons given above with respect to claim 1.

Claims 2-4 depend from claim 1. As explained in Section A, above, the Examiner’s rejection of claim 1 under 35 U.S.C. § 103(a) is not supported by Handler et al. and Basin et al., whether taken alone or in combination. Therefore, the

Examiner's rejections of claims 2-4 under 35 U.S.C. § 103(a) are likewise not supported by Hundler et al. and Basin et al. for at least the same reasons given above with respect to claim 1. Accordingly, the rejections of claims 2-4 must be reversed and the claims allowed.

C. **The rejection of independent claims 5 and 11 under 35 U.S.C. § 103(a) must be reversed because Hundler et al. and Basin et al. fail to support the rejection of these claims, which recite, among other things, a receiving module "for receiving a streamed un-extracted zip file," and an interface module "for accessing contents of a central directory of the streamed zip file as the central directory is received."**

In the rejections of claims 5 and 11, the Examiner asserts that Handler et al. teaches "a receiving module ... for receiving a streamed un-extracted zip file" as the receiving device 410 includes a streaming execut[o]r 416 that controls the receipt of the streamed modules." Final Office Action, p. 5, § 9, II. 10-13 (citing Handler et al., FIGS. 4 and 6; and ¶ 0075); see also id., p. 8, II. 3-6. However, the Examiner admits that Handler et al. fails to teach "receiving an un-extracted zip file" See id., p. 6, l. 1, and p. 8, II. 6-7. Therefore, contrary to the Examiner's assertions, Handler et al. fails to teach "a receiving module ... for receiving a streamed *un-extracted* zip file." Claim 5, II. 6-7 (emphasis added); see also claim 11, l. 5.

In an attempt to cure this deficiency, the Examiner again cites Basin et al. as teaching the existence of an un-extracted zip file, and asserts that it would have been obvious to the artisan to combine Handler et al. and Basin et al. to arrive at the claimed invention. Final Office Action, p. 6, II. 9-11; id., p. 8, II. 13-17. Appellants respectfully disagree.

The Examiner asserts that Hundler et al. teaches “an interface module, initiated by the application program, for accessing contents of the streamed zip file as it is received’ as the integration of streamed modules with executing modules are provided by client 410 dynamic module linking facilities.” Id., p. 6, II. 11-15 (citing Hundler et al., FIG. 4; and ¶ 0065); see also id., p. 8, II. 17-21. However, the cited portions of Hundler et al. make no reference to “receiving a streamed *un-extracted* zip file.” Claim 5, II. 6-7 (emphasis added); claim 11, I. 5 (emphasis added). Instead, as explained in Section A, above, Hundler et al. teaches to stream zip files as a series of separate modules in *extracted* form. See Hundler et al., ¶ 0073, II. 2-7. Thus, Hundler et al. fails to teach “an interface module … for accessing contents of a central directory of the streamed zip file as *the central directory is received*.” Claim 5, II. 8-10 (emphasis added); claim 11, II. 6-7 (emphasis added).

Moreover, Basin et al. is not relied upon to teach, and does not teach, the claimed interface module. As explained in Section A, above, Basin et al. provides access to the central directory only *after* the entire zip file has been received and extracted by the user. Basin et al., ¶ 0032, II. 6-9; ¶ 0037, II. 1-8. Consequently, even if the artisan would have been motivated to modify Hundler et al. by providing a receiving module for receiving a streamed un-extracted zip file (which Appellants dispute), neither Hundler et al., nor Basin et al., nor their combination, teach “an interface module … for accessing contents of a central directory of the streamed zip file as *the central directory is received*.”

For at least these reasons, Hundler et al. and Basin et al. fail to support the Examiner’s rejection of claims 5 and 11. Accordingly, Appellants respectfully request

that the rejection of claims 5 and 11 under 35 U.S.C. § 103(a) be reversed and the claims allowed.

- D. **The rejection of dependent claims 6 and 12 under 35 U.S.C. § 103(a) must be reversed because Hendler et al. and Basin et al. fail to teach that “the interface module is a Java class comprising a central header subclass and a central directory subclass.”**

Claim 6 and 12 depend from claims 5 and 11, respectively. As explained in Section C, above, the Examiner’s rejections of claims 5 and 11 under 35 U.S.C. § 103(a) lack support in Hendler et al. and Basin et al., whether taken singly or in combination. Therefore, the Examiner’s rejections of claims 6 and 12 likewise lack support in Hendler et al. and Basin et al. for at least the same reasons given above with respect to claims 5 and 11. Accordingly, the rejection of claims 6 and 12 under 35 U.S.C. § 103(a) must be reversed and the claims allowed.

In addition, in the rejections of claims 6 and 12, the Examiner asserts that Hendler et al. teaches that “the interface module is a Java class comprising a central header subclass and a central directory subclass’ as the streamed modules have a Java class comprising central directory headers 641-646 and central directory 640 as subclasses.” Final Office Action, p. 6, § 10, II. 1-4 (citing Hendler et al., FIG. 6; and ¶ 0084. Appellants respectfully disagree with the Examiner’s characterization of Hendler et al..

Contrary to the Examiner’s assertions, the cited portions of Hendler et al. describe *the streamed file*, itself, and not the alleged interface. As disclosed by Hendler et al., “an *extracted file* may be further subdivided before streaming. For example, [manifest] *file 601* may be a class file containing several Java classes.

Individual classes may be extracted from the file 601 and streamed as separate modules." Hundler et al., ¶ 0084, II. 1-5. Nowhere does Handler et al. teach an interface module "for accessing contents of a central directory of the streamed zip file as the central directory is received," as recited in claims 5 and 11, "wherein *the interface module* is a Java class comprising a central header subclass and a central directory subclass," as recited in claims 6 and 12 (emphasis added). Moreover, Basin et al. is not relied upon to teach, and, in fact, does not teach the cited deficiencies of Handler et al..

For at least these additional reasons, the Examiner's rejection of claims 6 and 12 lacks support in Handler et al. and Basin et al., whether taken alone or in combination. Accordingly, Appellants respectfully request that the rejection of claims 6 and 12 be withdrawn and the claims allowed.

E. The rejection of dependent claims 7-10 and 13-16 under 35 U.S.C. § 103(a) must be reversed for at least the same reasons given above with respect to claims 6 and 12.

Claims 7-10 depend from claim 6; claims 13-16 depend from claim 12. As explained in Section C, above, the Examiner's rejections of claims 5 and 11 under 35 U.S.C. § 103(a) lack support in Handler et al. and Basin et al., whether taken singly or in combination. Therefore, the Examiner's rejections of claims 7-10 and 13-16 under 35 U.S.C. § 103(a) likewise lack support in Handler et al. and Basin et al. for at least the same reasons given above with respect to claims 5 and 11. Accordingly, the rejections of claims 7-10 and 13-16 must be reversed and the claims allowed.

- F. The rejection of independent claim 17 under 35 U.S.C. § 103(a) must be reversed because Hundler et al. and Basin et al. fail to support the rejection of claim 17, which recites, among other things, “an interface ... for use with a receiver configured for receiving a streamed un-extracted zip file, wherein the zip file comprises a set of files and a central directory, the interface comprising a process for accessing contents of the central directory as it is received.”

In the rejection of claim 17, the Examiner asserts that Handler et al. teaches

“an interface stored in [a] memory, the interface for use with a receiver configured for receiving a streamed ... un-extracted zip file, wherein the zip file comprises a set of files and a central directory, the interface comprising a process for accessing contents of the central directory as it is received” as the receiving device 410 includes a streaming execut[o]r 416 that controls the receipt of the streamed modules.

Final Office Action, p. 10, § 21, II. 5-11 (citing Handler et al., FIGS. 4 and 6; and ¶ 0075). The Examiner also cites Basin et al. as teaching the existence of an un-extracted zip file, and asserts that it would have been obvious to the artisan to combine Handler et al. and Basin et al. to arrive at the claimed invention. Id., § 21, II. 13, *et seq.* However, Appellants respectfully disagree with the Examiner’s characterization of the cited references.

As the Examiner admits (id., § 21, II. 13-14), Handler et al. fails to teach “receiving an un-extracted zip file.” Therefore, contrary to the Examiner’s assertions, Handler et al. fails to teach “an interface for use with a receiver configured to receive a streamed un-extracted zip file.” Claim 17, II. 3-4. As explained in Section A, above, Handler et al. teaches to extract the zip file before streaming its constituent files. See Handler et al., ¶¶ 0073-76. Therefore, Handler et al. fails to teach “a process for

accessing contents of [a] central directory as *it is received.*" Claim 17, II. 5-6 (emphasis added).

Moreover, Basin et al. is not relied upon to teach, and does not teach, "a process for accessing contents of the central directory as *it is received.*" Claim 17, II. 5-6 (emphasis added). Instead, the Basin et al. method allows the central directory to be accessed only *after* the entire zip file has been received and extracted by the user. Basin et al., ¶ 0032, II. 6-9, ¶ 0037, II. 1-8.

Consequently, even if the artisan would have been motivated to modify Handler et al. by streaming the JAR file in un-extracted form (which Appellants dispute), neither Handler et al., nor Basin et al., nor their combination, teaches "an interface ... for use with a receiver configured for receiving a streamed un-extracted zip file, wherein the zip file comprises a set of files and a central directory, the interface comprising a process for accessing contents of [a] central directory as it is received."

For at least these reasons, Handler et al. and Basin et al. fail to support the Examiner's rejection of claim 17, whether taken alone or in combination. Accordingly, Appellants respectfully request that the rejection of claim 17 under 35 U.S.C. § 103(a) be reversed and the claim allowed.

G. The rejection of dependent claims 18 and 19 under 35 U.S.C. § 103(a) must be reversed for at least the same reasons given above with respect to claim 17.

Claims 18 and 19 depend from claim 17. As explained in Section F, above, the Examiner's rejection of claim 17 under 35 U.S.C. § 103(a) is not supported by Handler et al. and Basin et al., whether taken alone or in combination. Therefore, the

Examiner's rejections of claims 18 and 19 are likewise not supported by Handler et al. and Basin et al. for at least the same reasons given above with respect to claim 17. Accordingly, the rejections of claims 18 and 19 under 35 U.S.C. § 103(a) must be reversed and the claims allowed.

H. Conclusion

For at least the reasons given above, the Examiner's rejection of pending claims 1-19 lacks support in Handler et al. and Basin et al., whether taken alone or in combination. Accordingly, Appellants respectfully request that the Examiner's rejections of claims 1-19 under 35 U.S.C. § 103(a) be reversed and the claims allowed.

To the extent any extension of time under 37 C.F.R. § 1.136 is required to obtain entry of this Appeal Brief, such extension is hereby respectfully requested. If there are any fees due under 37 C.F.R. §§ 1.16 or 1.17 which are not enclosed herewith, including any fees required for an extension of time under 37 C.F.R. § 1.136, please charge such fees to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: December 29, 2005

By: _____

John M. Mulcahy
Reg. No. 55,940



Application No.: 10/051,277
Attorney Docket No.: 06502.0396-00

VIII. CLAIMS APPENDIX

1. A method of accessing a streamed zip file comprising:
receiving a stream of data containing an un-extracted zip file, wherein the zip file comprises a set of files and a central directory; and
enabling a process to access contents of the central directory as it is received.

2. The method of claim 1 wherein the enabling step comprises providing an interface for accessing contents of the central directory.

3. The method of claim 1 wherein the enabling step comprises:
reading in a central directory file header; and
providing an interface to access contents of the central directory file header.

4. The method of claim 1 wherein the enabling step comprises:
reading in an end of central directory record; and
providing an interface to access contents of the end of central directory record.

5. A system for receiving a streamed zip file, wherein the zip file comprises a set of files and a central directory, the system comprising:

a central processing unit;
an application program configured for execution by the central processing unit;
a receiving module, initiated by the application program, for receiving a streamed un-extracted zip file; and
an interface module, initiated by the application program, for accessing contents of a central directory of the streamed zip file as the central directory is received.

6. The system of claim 5 wherein the interface module is a Java class comprising a central header subclass and a central directory subclass.

7. The system of claim 6 wherein the receiver module reads in a central directory header as an object instance of the central header subclass.

8. The system of claim 7 wherein the central header subclass comprises a set of methods for accessing contents of the object instance.

9. The system of claim 6 wherein the receiver module reads in an end of central directory record as an object instance of the central directory subclass.

10. The system of claim 9 wherein the central directory subclass comprises a set of methods for accessing contents of the object instance.

11. A computer program product for use in conjunction with a computer system, the computer program product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising:

a receiving module for receiving a streamed un-extracted zip file; and
an interface module for accessing contents of a central directory of the streamed zip file as the central directory is received.

12. The computer program mechanism of claim 11 wherein the interface module is a Java class comprising a central header subclass and a central directory subclass.

13. The computer program mechanism of claim 12 wherein the receiver module reads in a central directory header as an object instance of the central header subclass.

14. The computer program mechanism of claim 13 wherein the central header subclass comprises a set of methods for accessing contents of the object instance.

15. The computer program mechanism of claim 12 wherein the receiver module reads in an end of central directory record as an object instance of the central directory subclass.

16. The computer program mechanism of claim 15 wherein the central directory subclass comprises a set of methods for accessing contents of the object instance.

17. A memory for storing data for access by an application program being executed on a computer system, comprising:

an interface stored in the memory, the interface for use with a receiver configured for receiving a streamed un-extracted zip file, wherein the zip file comprises a set of files and a central directory, the interface comprising a process for accessing contents of the central directory as it is received.

18. The memory of claim 17 wherein the process is a Java class comprising a set of methods for accessing contents of an object instance of the Java class, wherein the object instance comprises a central directory header read in by the receiver.

19. The memory of claim 17 wherein the process is a Java class comprising a set of methods for accessing contents of an object instance of the Java class, wherein the object instance comprises an end of central directory record read in by the receiver.

IX. EVIDENCE APPENDIX

None.

X. **RELATED PROCEEDINGS APPENDIX**

None.